

# WEST Search History

DATE: Tuesday, November 18, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
	<i>DB=EPAB; PLUR=YES; OP=ADJ</i>		
L18	L13	2	L18
	<i>DB=USPT; PLUR=YES; OP=ADJ</i>		
L17	L16	22	L17
	<i>DB=USPT,PGPB; PLUR=YES; OP=ADJ</i>		
L16	L8 and (HTML or DHTML or SGML or XML or bookmaster)	44	L16
	<i>DB=JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>		
L15	L13 and (HTML or DHTML or SGML or XML or bookmaster)	0	L15
L14	L13 and (tag or tagging or tagged)	2	L14
L13	test plan or test?plan or testplan	66	L13
	<i>DB=USPT,PGPB; PLUR=YES; OP=ADJ</i>		
L12	(L8 or l9) and l7	30	L12
L11	L8 and tagging	8	L11
L10	L8 and tag	52	L10
L9	L8 and tag\$	47	L9
L8	test plan or test?plan or testplan	371	L8
L7	L6 or l5 or l4 or l3 or l2 or l1	4809	L7
L6	((715/906  715/907  715/908  715/909  715/910  715/911 )!.CCLS. )	41	L6
L5	((715/513  715/523  715/524  715/536  715/537  715/538  715/539  715/540  715/541 )!.CCLS. )	1692	L5
L4	((712/226  712/227 )!.CCLS. )	650	L4
L3	((703/14  703/15 )!.CCLS. )	761	L3
L2	((702/118  702/119  702/120 )!.CCLS. )	391	L2
L1	((717/113  717/117  717/123  717/124  717/125  717/126  717/127  717/128  717/131  717/135  717/136  717/137 )!.CCLS. )	1353	L1

END OF SEARCH HISTORY

**Generate Collection****Print****Search Results - Record(s) 1 through 20 of 30 returned.** **1. Document ID: US 20030212523 A1**

L12: Entry 1 of 30

File: PGPB

Nov 13, 2003

PGPUB-DOCUMENT-NUMBER: 20030212523

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030212523 A1

TITLE: Dynamically adaptable semiconductor parametric testing

PUBLICATION-DATE: November 13, 2003

**INVENTOR-INFORMATION:**

NAME	CITY	STATE	COUNTRY	RULE-47
Dorough, Michael J.	Meridian	ID	US	
Blunn, Robert G.	Boise	ID	US	
Velichko, Sergey A.	Boise	ID	US	

US-CL-CURRENT: 702/119

**ABSTRACT:**

An apparatus, method, system, and signal-bearing medium may provide multiple maps, which may include multiple probing sequences to be called upon at run-time based on statistical thresholds or other selected criteria. Each map may include a series of locations on a wafer, the tests to perform at each location, and the measured results of each test. A parametric test system may perform the test at the associated location on the wafer. If the statistical threshold is exceeded or the selected criteria is met, the current map may be abandoned in favor of a different map.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [Claims](#) | [EPOC](#) | [Draw Desc](#) | [Image](#) **2. Document ID: US 20030196190 A1**

L12: Entry 2 of 30

File: PGPB

Oct 16, 2003

PGPUB-DOCUMENT-NUMBER: 20030196190

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030196190 A1

TITLE: Generating and managing test plans for testing computer software

PUBLICATION-DATE: October 16, 2003

**INVENTOR-INFORMATION:**

NAME	CITY	STATE	COUNTRY	RULE-47
Ruffolo, Nuzio	Markham		CA	
Chan, Keith	Markham		CA	
Cialini, Enzo	Mississauga		CA	
Di Loreto, Anthony	Markham		CA	

ABSTRACT:

An aspect of the present invention provides a system and a method for generating and managing test plans for guiding a test team through the process of testing computer software. Each component of computer software performs at least one specific task or function. A test plan includes several component test plans each for guiding the test team through the process of testing components of computer software. A component test plan includes a set of test cases or test scenarios. Each test case identifies items (that is, functional aspects of the computer software) for guiding the test team when they test a desired component of software. A distribution list is associated with at least one component of computer software. The distribution list identifies items related to the component of computer software. The distribution list also identifies the number of occurrences of each item in the component test plan (spread amongst several test cases of the component test plan). In a preferred embodiment, one test item is included per test case. Components of computer software and associated distribution lists are identified and subsequently the test plan is generated

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [Claims](#) | [KMC](#) | [Drawn Desc](#) | [Image](#)

---

3. Document ID: US 20030182097 A1

L12: Entry 3 of 30

File: PGPB

Sep 25, 2003

PGPUB-DOCUMENT-NUMBER: 20030182097

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030182097 A1

TITLE: Electronic device design-aiding apparatus, electronic device design-aiding method, electronic device manufacturing method, and computer readable medium storing program

PUBLICATION-DATE: September 25, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Furukawa, Yasuo	Tokyo		JP	

US-CL-CURRENT: 703/15

ABSTRACT:

During designing an electronic device, a test method and a peripheral circuit are also designed using logic data for simulating the operation of the electronic device and the characteristics of a test apparatus used for testing an electronic device. By using the designed test method and logic data representing the operation of the designed peripheral circuit, simulation to judge whether or not the electronic device can be tested. According to the results of the simulation, the designs of the electronic device, the test method, and the peripheral circuit are altered. To optimize the designs of the electronic device, the test method, and the peripheral circuit, simulation is repeated.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [KMC](#) | [Drawn Desc](#) | [Image](#)

---

4. Document ID: US 20030084429 A1

PGPUB-DOCUMENT-NUMBER: 20030084429  
PGPUB-FILING-TYPE: new  
DOCUMENT-IDENTIFIER: US 20030084429 A1

TITLE: Systems and methods for table driven automation testing of software programs

PUBLICATION-DATE: May 1, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Schaefer, James S.	Richmond	VA	US	

US-CL-CURRENT: 717/125; 717/127

ABSTRACT:

A table driven test automation system for performing functional testing of a software program. The system may include a GUI translator component to translate one or more GUI maps into a set of database tables, a data input component to facilitate entry and editing of test case data in the tables, and a test engine component for executing the software program based on a test case stored in the tables.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [EPOC](#) | [Draw Desc](#) | [Image](#)

---

5. Document ID: US 20030037314 A1

PGPUB-DOCUMENT-NUMBER: 20030037314  
PGPUB-FILING-TYPE: new  
DOCUMENT-IDENTIFIER: US 20030037314 A1

TITLE: Method and apparatus for testing and evaluating a software component using an abstraction matrix

PUBLICATION-DATE: February 20, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Apuzzo, Joseph T.	Staatsburg	NY	US	
Marino, John P.	Poughkeepsie	NY	US	
Hoskins, Curtis L.	Poughkeepsie	NY	US	
Race, Timothy L.	Kingston	NY	US	
Suri, Hemant R.	Poughkeepsie	NY	US	

US-CL-CURRENT: 717/125

ABSTRACT:

A functional testing and evaluation technique is provided employing an abstraction matrix that describes a complex software component to be tested. The abstraction matrix includes at least one test case scenario and mapped expected results therefore. Test cases are derived from the at least one test case scenario and used to test the software component, thereby generating test results. The test results are automatically evaluated using the abstraction matrix. The evaluating includes comparing a test case to the at least one test case scenario of the abstraction matrix and if a match is found, comparing the test result for that test case with the mapped expected result therefore in the abstraction matrix.

**□ 6. Document ID: US 20030033591 A1**

L12: Entry 6 of 30

File: PGPB

Feb 13, 2003

PGPUB-DOCUMENT-NUMBER: 20030033591

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030033591 A1

TITLE: Tool for implementing floating-point related applications using customized language

PUBLICATION-DATE: February 13, 2003

## INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Fournier, Laurent	Givat Elah		IL	

US-CL-CURRENT: 717/124; 717/115

## ABSTRACT:

A tool for implementing a Floating-Point related application. The tool includes a receiver for receiving a list of commands in a computer language. The language defines Floating-Point events of interest and the regrouping of events into a coverage model in respect of a desired FP instruction. The coverage model has the form of a sequence of Floating-Point commands with constraints on the input operands, intermediate result operand and the result operand. The constraints are expressed in terms of sets that define allowable Floating-Point numbers. The tool further includes a parser for parsing the commands and a processor for processing the parsed commands for realizing on the basis of the events and the coverage model the Floating-point related application.

**□ 7. Document ID: US 20030028856 A1**

L12: Entry 7 of 30

File: PGPB

Feb 6, 2003

PGPUB-DOCUMENT-NUMBER: 20030028856

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030028856 A1

TITLE: Method and apparatus for testing a software component using an abstraction matrix

PUBLICATION-DATE: February 6, 2003

## INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Apuzzo, Joseph T.	Staatsburg	NY	US	
Marino, John P.	Poughkeepsie	NY	US	
Hoskins, Curtis L.	Poughkeepsie	NY	US	
Race, Timothy L.	Kingston	NY	US	
Suri, Hemant R.	Poughkeepsie	NY	US	

US-CL-CURRENT: 717/124; 714/38

**ABSTRACT:**

A functional testing technique is provided employing an abstraction matrix that describes a complex software component to be tested. The abstraction matrix includes state and event information. The technique is an automated process which parses the abstraction matrix to generate test cases and mapped expected results therefore. The test cases are separated based on layers of the software component and data structures are associated with the separated test cases of the layers. The data structures allow the test cases of the various layers to be uncorrelated. The software component executable is employed to generate test case execution threads from the test cases and mapped expected results for a particular layer. These execution threads can then be executed in parallel, thereby testing the software component.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Reviews](#) | [Classification](#) | [Date](#) | [Preference](#) | [Sequences](#) | [Attachments](#)

[HTML](#) | [Drawn Desc](#) | [Image](#)

---

8. Document ID: US 20030014734 A1

L12: Entry 8 of 30

File: PGPB

Jan 16, 2003

PGPUB-DOCUMENT-NUMBER: 20030014734

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030014734 A1

TITLE: Technique using persistent foci for finite state machine based software test generation

PUBLICATION-DATE: January 16, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Hartman, Alan	Haifa	MA	IL	
Hagin, Kenneth	Nazareth Illit		IL	
Kram, Paul	Lowell		US	

US-CL-CURRENT: 717/125

ABSTRACT:

A system for automatic generation of test programs employs test generation foci linked to a finite state machine behavioral model of a software application under test to produce abstract test suites that are executed by an execution engine. The foci include directive expressions that tag coverage variables of the behavioral model. The execution engine operates until the tagged coverage variables have assumed all their possible values. Both the behavioral models and the foci are archivable and can be retrieved and reused independently.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Reviews](#) | [Classification](#) | [Date](#) | [Preference](#) | [Sequences](#) | [Attachments](#)

[HTML](#) | [Drawn Desc](#) | [Image](#)

---

9. Document ID: US 20020049962 A1

L12: Entry 9 of 30

File: PGPB

Apr 25, 2002

PGPUB-DOCUMENT-NUMBER: 20020049962

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020049962 A1

TITLE: Product testing and bug tracking system

PUBLICATION-DATE: April 25, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Kelbaugh, Michael	Kirkland	WA	US	
Diederich, Dave	Kirkland	WA	US	
Bush, Eric M.	Monroe	WA	US	

US-CL-CURRENT: 717/128

ABSTRACT:

An Internet-based, secure communications system is utilized for enabling communications between a video game tester, project coordinator and others with a game developer. A master bug log which compiles all uncovered bugs is accessible by a game developer and other authorized system users via a web server, which stores bug tracking system applications programs and associated data bases. Such a master bug log includes a file attachment capability permitting a digitized image file replicating a video game display screen sequence depicting the bug, to be attached for downloading to, for example, a game developer. Bugs may be sorted, for example, so that a game developer can retrieve only those bugs having a digitized file attachment. Sorting may take place based on any of a large number of fields entered in the master bug log. The present exemplary embodiments permit customized fields to be added and used as sort criteria. For example, in a racing game, bugs may be categorized and sorted based upon involvement with a particular vehicle or driver. Game and debugging related messages may be exchanged between testers, project coordinators, and corporate contacts. If the game developer normally communicates in, for example, Japanese, e-mail type format messages are translated so that significant game related messages may be promptly analyzed by all parties involved. An editing function is advantageously utilized to permit, for example, a tester to enter a bug description and a project coordinator to edit the tester's description. The illustrative embodiments of the present invention advantageously use multiple security layers to preclude one developer from accessing information related to a game under test developed by another developer.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [RMC](#) | [Draw Desc](#) | [Image](#)

---

10. Document ID: US 20020042687 A1

L12: Entry 10 of 30

File: PGPB

Apr 11, 2002

PGPUB-DOCUMENT-NUMBER: 20020042687

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020042687 A1

TITLE: System, method and medium for certifying and accrediting requirements compliance

PUBLICATION-DATE: April 11, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Tracy, Richard P.	Ashburn	VA	US	
Barrett, Hugh	Centreville	VA	US	
Berman, Lon J.	Sterling	VA	US	
Catlin, Gary M.	Bricktown	NJ	US	

US-CL-CURRENT: 702/119; 702/123

ABSTRACT:

A computer-implemented system, method and medium for assessing the risk of and/or determining the suitability of a system to comply with at least one predefined standard, regulation and/or requirement. In at least some embodiments of the present invention, the method comprises the steps of: 1) automatically or manually gathering information pertaining to the system, 2) selecting one or more requirements with which the system is to comply; 3) testing the system against the requirements; 4) performing risk assessment of the failed test procedures, and 5) generating certification documentation based on an assessment of the first four elements.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [RWC](#) | [Drawn Desc](#) | [Image](#)

---

11. Document ID: US 20020029377 A1

L12: Entry 11 of 30

File: PGPB

Mar 7, 2002

PGPUB-DOCUMENT-NUMBER: 20020029377

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020029377 A1

TITLE: System and method for developing test cases using a test object library

PUBLICATION-DATE: March 7, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Pavela, Thomas J.	San Jose	CA	US	

US-CL-CURRENT: 717/124

ABSTRACT:

A method, apparatus, article of manufacture, and a memory structure for generating a test code for an automatic procedure is disclosed. The method comprises the steps of defining a source file having a plurality of tags associated with a member of a library of executable code objects defining a set of instructions for performing a portion of the automatic test procedure, generating a test plan in a conventional language from the source file, and generating an automated test code for the automated test procedure from the source file. In one embodiment, a test index identifying system elements tested by the test code is generated and incorporated into the test plan, allowing the user to verify that all desired system elements are exercised by the automated test code. The article of manufacture comprises a data storage device tangibly embodying instructions to perform the method steps described above. The apparatus comprises means for defining a source file having a plurality of tags, wherein each tag is associated with a member of a library of executable code objects defining a set of instructions for performing a portion of an automatic test procedure, means for generating a test plan in a conversational language from the source file, and means for generating an automated test code for the automatic test procedure from the source file.

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [RWC](#) | [Drawn Desc](#) | [Image](#)

---

12. Document ID: US 6643555 B1

L12: Entry 12 of 30

File: USPT

Nov 4, 2003

US-PAT-NO: 6643555

DOCUMENT-IDENTIFIER: US 6643555 B1

TITLE: Method and apparatus for generating an application for an automation control system

DATE-ISSUED: November 4, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Eller; Thomas	Moerfelden			DE
Peyrou; M. Remi	Frankfurt am Main			DE

US-CL-CURRENT: 700/83; 700/17, 700/18, 700/19, 700/23, 700/86, 700/87, 703/13,  
703/14, 703/15, 706/14, 706/21, 706/5, 706/920, 709/220, 709/221, 709/223, 718/106,  
718/107

ABSTRACT:

An apparatus and method of generating an application for a control system. A control process is defined by a physical model and a topological model. An application generator utilizes the physical and topological models to generate an application for the control system.

25 Claims, 29 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 29

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) |

[EPOC](#) | [Draw Desc](#) | [Image](#)

---

13. Document ID: US 6546506 B1

L12: Entry 13 of 30

File: USPT

Apr 8, 2003

US-PAT-NO: 6546506

DOCUMENT-IDENTIFIER: US 6546506 B1

TITLE: Technique for automatically generating a software test plan

DATE-ISSUED: April 8, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lewis; Erika B.	Cary	NC		

US-CL-CURRENT: 714/38; 717/101, 717/124

ABSTRACT:

A method, system, and computer program product for automatically generating a software test plan. The computation of test duration is based on tasks performed by test personnel, in particular the tasks of executing test scenarios and identifying (and documenting) defects, the number of hours spent on these tasks, and the number of test personnel available.

15 Claims, 2 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 2

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) |

[EPOC](#) | [Draw Desc](#) | [Image](#)

14. Document ID: US 6522987 B1

L12: Entry 14 of 30

File: USPT

Feb 18, 2003

US-PAT-NO: 6522987

DOCUMENT-IDENTIFIER: US 6522987 B1

TITLE: Monitoring system and method implementing a percent availability test

DATE-ISSUED: February 18, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Flink; Eric N.	Loveland	CO		
Chamberlain; Craig	Louisville	CO		
Morgan; Bill	Santa Rosa	CA		

US-CL-CURRENT: 702/122; 702/107, 702/120, 702/121, 702/126

ABSTRACT:

A channel plan with a corresponding test plan are implemented in connection with a plurality of nodes that communicate signals. The channel plan has one or more predefined specifications for each of one or more signal channels on each of the nodes. The channel plan enables a monitoring system to, among other things, conduct automatic periodic test plans, comprising tests, on the nodes, based upon the predefined data specified in the channel plan. Each test plan prescribes measurement of at least one signal parameter, pertaining to one or more nodes as a whole and/or to one or more channels contained within the nodes. The monitoring system includes a spectrum analyzer, a switch enabling the spectrum analyzer to interface with the nodes, and a controller controlling the switch and the spectrum analyzer. The controller is configured to enable creation of and display the channel plan and test plan, based upon user inputs. Notably, the controller can be configured to perform a percent availability test, that, when used in connection with an active channel, indicates how successful the system under test is in providing a service over the channel. When performed in connection with an unused channel, then the percent availability test indicates how successful the channel would be in providing a future service over the unused channel.

20 Claims, 59 Drawing figures

Exemplary Claim Number: 15

Number of Drawing Sheets: 59

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) |

[NONE](#) | [Draw Desc](#) | [Image](#) |

---

15. Document ID: US 6425118 B1

L12: Entry 15 of 30

File: USPT

Jul 23, 2002

US-PAT-NO: 6425118

DOCUMENT-IDENTIFIER: US 6425118 B1

TITLE: System for automatically generating tests to ensure binary compatibility between software components produced by a source-to-source computer language translator

DATE-ISSUED: July 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Molloy; Mark E.	Knoxville	IA		
Andrews; Kristy A.	Palo Alto	CA		
Herren; James L.	Milpitas	CA		
Cutler; David R.	San Jose	CA		
Del Vigna; Paul	San Jose	CA		

US-CL-CURRENT: 717/136; 717/162

ABSTRACT:

A system and method that implement a strategy for automatically generating self-checking tests of source-to-source translation. The tests make certain that software components produced by a source-to-source computer language translator continue to be binary-compatible with all other software components with which they previously interacted correctly, including components that remain in the original programming language. To that end, the correctness of procedure interfaces in a target computer program is also verified. With this strategy, correct translation of text preprocessor mechanisms such as macros, conditionally compiled regions of code, and source file inclusion can be achieved. The target program is the result of translating a source program written in a different language from the target program. The system creates a set of procedure calls to each procedure in the source program. The system also creates a set of callable procedures with the same interface specification as in the source program. However, the callable procedures created by the system to serve to perform tests on variables and other data that is passed to each procedures' interface specification. The system-generated procedure calls are designed to assign different test values to ensure that a valid interface exists. For example, high, low, zero, and random values, or other values, are chosen for a variable to test. The procedure calls, and callable procedures, are generated in the source language and are also translated into the target language by a translator. The procedure calls and callable procedures in both language formats are then cross-linked and executed to determine whether the translator is translating correctly with respect to the procedure interface specifications.

20 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [NAME](#) | [Draw Disc](#) | [Image](#)

16. Document ID: US 6421822 B1

L12: Entry 16 of 30

File: USPT

Jul 16, 2002

US-PAT-NO: 6421822

DOCUMENT-IDENTIFIER: US 6421822 B1

TITLE: Graphical user interface for developing test cases using a test object library

DATE-ISSUED: July 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pavela; Thomas J.	San Jose	CA		

US-CL-CURRENT: 717/125; 345/771, 345/970, 717/109

ABSTRACT:

A method, apparatus, article of manufacture, for generating a test code for an automatic procedure is disclosed. The method comprises the steps of presenting a

visual representation of a library of executable code objects comprising a plurality of test objects to a user, accepting a selection of a first test object in the visual representation, presenting first test object options defining at least one test parameter, accepting a selection of a first test option, translating the first test option into at least one tag and at least one tag parameter, and storing the tag and the tag parameter in a source file. The article of manufacture comprises a data storage device tangibly embodying instructions to perform the method steps described above. The apparatus comprises computer with suitable program instructions for presenting a visual representation of a library of executable code objects to a user. The library of executable code objects includes a plurality of test object members, each of which define a set of instructions for performing a portion of the test procedure. The computer is also configured to implement a graphical user interface, to accepting a selection of a first test object in the visual representation, to present test options for the selected test object, and to accept a test object option.

21 Claims, 40 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 40

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#)

[KINIC](#) | [Draw Desc](#) | [Image](#)

17. Document ID: US 6418391 B1

L12: Entry 17 of 30

File: USPT

Jul 9, 2002

US-PAT-NO: 6418391

DOCUMENT-IDENTIFIER: US 6418391 B1

TITLE: Testing system for performing an operation of an application which controls testing equipment for testing a device under test and method for controlling the same

DATE-ISSUED: July 9, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Umezu; Satoshi	Tokyo			JP
Yamaguchi; Takahiro	Tokyo			JP
Miyajima; Jun	Tokyo			JP

US-CL-CURRENT: 702/123; 702/119, 702/120

ABSTRACT:

There is disclosed a testing system for performing an operation of an application which controls testing equipment for testing a device under test by displaying images such as icons or buttons on a screen, and selecting these images with use of a pointing device, and a method for controlling the testing system. An icon corresponding to the device under test and an icon corresponding to a test element are displayed, and then connected and displayed. In addition, a test parameter is generated corresponding to each test element, a test is conducted by setting respective test parameters, and obtained test data is displayed.

12 Claims, 25 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 24

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#)

[KINIC](#) | [Draw Desc](#) | [Image](#)

18. Document ID: US 6405364 B1

L12: Entry 18 of 30

File: USPT

Jun 11, 2002

US-PAT-NO: 6405364  
DOCUMENT-IDENTIFIER: US 6405364 B1

TITLE: Building techniques in a development architecture framework

DATE-ISSUED: June 11, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/101; 717/102, 717/120, 717/124

ABSTRACT:

A system is provided for building systems in a development architecture framework. The present invention is directed to both a system to be built and an implementation strategy to fulfill system requirements. Software components of the system are encapsulated with wrappers. The wrappers are adapted to be changed upon other software components of the system being changed while the encapsulated software components of the system remain unchanged. In one embodiment of the present invention, specifying the requirements of the system to be built and the implementation strategy to fulfill the requirements may be carried out using tools such as data modeling tools, process modeling tools, event modeling tools, performance modeling tools, object modeling tools, component modeling tools, reuse support tools, prototyping tools, application logic design tools, database design tools, presentation design tools, communication design, and usability test tools. In another embodiment of the present invention, improving the performance and maintenance of the system may be carried out using tools such as interactive navigation tools, graphical representation tools, extraction tools, repository tools, restructuring tools, and data name rationalization tools.

12 Claims, 14 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 14

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#)

[KMM](#) | [Drawn Dese](#) | [Image](#)

---

19. Document ID: US 6332211 B1

L12: Entry 19 of 30

File: USPT

Dec 18, 2001

US-PAT-NO: 6332211  
DOCUMENT-IDENTIFIER: US 6332211 B1

TITLE: System and method for developing test cases using a test object library

DATE-ISSUED: December 18, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pavela; Thomas J.	San Jose	CA		

US-CL-CURRENT: 717/130; 702/119

ABSTRACT:

A method, apparatus, article of manufacture, and a memory structure for generating a

test code for an automatic procedure is disclosed. The method comprises the steps of defining a source file having a plurality of tags associated with a member of a library of executable code objects defining a set of instructions for performing a portion of the automatic test procedure, generating a test plan in a conventional language from the source file, and generating an automated test code for the automated test procedure from the source file. In one embodiment, a test index identifying system elements tested by the test code is generated and incorporated into the test plan, allowing the user to verify that all desired system elements are exercised by the automated test code. The article of manufacture comprises a data storage device tangibly embodying instructions to perform the method steps described above. The apparatus comprises means for defining a source file having a plurality of tags, wherein each tag is associated with a member of a library of executable code objects defining a set of instructions for performing a portion of an automatic test procedure, means for generating a test plan in a conversational language from the source file, and means for generating an automated test code for the automatic test procedure from the source file.

18 Claims, 40 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 40

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [KIMC](#) | [Draw Desc](#) | [Image](#)

---

20. Document ID: US 6226784 B1

L12: Entry 20 of 30

File: USPT

May 1, 2001

US-PAT-NO: 6226784

DOCUMENT-IDENTIFIER: US 6226784 B1

TITLE: Reliable and repeatable process for specifying developing distributing and monitoring a software system in a dynamic environment

DATE-ISSUED: May 1, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Holmes; Marty	Suwanee	GA		
Jan; Imran	Duluth	GA		
Lockie; Kenneth	Alpharetta	GA		
McGuire; Kimberley L.	Atlanta	GA		

US-CL-CURRENT: 717/100, 717/124, 717/176, 717/177

ABSTRACT:

A reliable and repeatable process for specifying, developing, distributing, and monitoring a software system or application within a dynamic environment includes the steps of specifying a set of software system requirements during a discovery process within a data processing environment, establishing a development and delivery schedule for distribution of the software system, developing a software system corresponding to the software system requirements during a development process, and developing and testing an install package to be used to install the software system in a test environment within the data processing environment. The install package includes the software system and routines configured to automatically instantiate the software system in the test environment. The developing and testing step takes a first amount of time. Other steps include distributing said install package to a test user community for testing of said software system during a testing process which takes a second amount of time, and revising the software system to comply with the software system requirements when a defect is discovered and then repeating the developing and testing and the distributing steps. The revising step affects the delivery schedule by adding a predetermined amount of time thereto. The process also includes a step of placing

the software system or the revised software system into general use after successful completion of the testing process. The placing step takes a third amount of time. The first, second, and third amounts of time are pre-determined based on the development and delivery schedule prior to development of the software system.

11 Claims, 4 Drawing figures  
Exemplary Claim Number: 1  
Number of Drawing Sheets: 4

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [HOME](#) | [Draw Desc](#) | [Image](#)

[Generate Collection](#)

[Print](#)

<b>Terms</b>	<b>Documents</b>
(L8 or l9) and l7	30

**Display Format:**  [Change Format](#)

[Previous Page](#)      [Next Page](#)